2022 ASABE Robotics Competition

Robotics Written Report

Team Name:	Tensor
Division:	Advanced
University:	Zhejiang University
Advisors:	Yibin YING, Huanyu JIANG, Zunzhong YE, Di CUI, Yonghua YU
Team Captain:	Yiyuan LIN
Email:	yiyuanlin@zju.edu.cn
Team members:	Xin JIANG, Yuliang ZHOU, Xinyue SUN, Yilin ZHANG, Zenan SHENG, Hangyu ZHOU

Catalog

1 Information About Team Members

Table 1-1 Profile of team members

Name	Assigned tasks	Note
Yiyuan Lin	Navigation and Motion Planning	Captain Undergraduate Junior
Xin Jiang	Manipulator Control	Undergraduate Junior
Yuliang Zhou	Mechanical Structure Design	Undergraduate Junior
Yilin Zhang	Overall Design and Debugging	Undergraduate Junior
Hangyu Zhou	Machine Vision	Undergraduate Senior
Xinyue Sun	Circuit Design	Undergraduate Junior
Zenan Sheng	Structure Design	Undergraduate Sophomore

2 Establishment of Need and Benefit to Agriculture

2.1 Cotton Plant Overview

Cotton is the world's most prevalent and profitable non-food crop. It is a shrub native to tropical and subtropical regions around the world, including the Americas, Africa, Egypt, and India. Current estimates for world production are about 25 million tonnes or 110 million bales annually, accounting for 2.5% of the world's arable land. Its production provides income for more than 250 million people worldwide.

Conventional hand-picked cotton quality is stable, and the market welcomes it, but due to the short picking period, labor intensity, shortage of workers. Mechanized picking is the direction of development, but the cost is too high, and quality is not guaranteed without manual verification.

Nowadays, cotton in more and more countries and regions is harvested mechanically, either by a cotton picker, a machine that removes the cotton bolls without damaging the cotton plant, or by a cotton stripper, which strips the entire boll off the plant.

2.2 Design Meaning

2.2.1 Adaptable to the Diversity of Cotton Varieties

The agricultural robot simulates manual real-time picking with programmability, adaptability, and versatility, and its flexible design features can solve the problems of many cotton species and long maturity periods.

2.2.2 Low Agronomic Requirements

Agricultural robots can mimic the precise movements of manual picking, with a large degree of spatial and temporal freedom of picking, which can overcome the shortcomings of mechanical cotton picking with high agronomic requirements, avoiding leaf debris pollution and saving the cost of chemical agents, while ensuring the natural growth of cotton crops and contributing to soil protection.

2.2.3 Improve the Yield of High-quality Cotton

Agricultural robots with their intelligent automatic grading technology while picking, and grading can improve the quality of pre-harvest cotton grading and the yield of high-quality cotton. In the United States, due to the uniform cotton species, short maturity span and a unified operation, the quality of cotton in the machine-picked bales is comparable, and the sampling and grading inspection results of the bales are of commercial value. In China, due to the many cotton species, long maturity, and batch-harvesting, traditional picking is bound to confuse the grade of cotton in the bale, cotton bale sampling, and grading test results do not have a price orientation, and agricultural robots can be a good solution to this problem.

2.2.4 Reduce Production Costs

The lack of labor for cotton picking and the short cotton mature period with high labor intensity determine the development direction of mechanical picking. Agricultural robots simulate manual cotton picking, which can ensure the quality of picking, and at the same time, it has the characteristics of small size, flexible movement, and easy handling over long distances, and can also play the role of other agricultural production tools, which can be used in one machine.

3. Definition of Design Objectives and Criteria

3.1 Design Objectives

The overall design goal of the cotton-picking robot is to complete the identification of opened and unopened cotton bolls on simulated cotton plants throughout the venue, followed by nondestructive picking and collection of opened bolls and mapping and display of unopened bolls. The objectives are now split as follows.

A. Implement robot autonomous navigation
B. Location of the cotton plant
C. Identify the state of cotton boll
D. Mapping and displaying unopened bolls on the LCD panel
E. Pick the open bolls
F. Load the harvested cotton bolls into the hopper truck

3.2 Design Criteria

Performing tasks effectively in a simple, stable, low-cost, and efficient manner.

(1) Simple, reliable, and low-cost hardware

We want to make as few and as simple devices as possible to control identification and picking. We believe that simple, reliable, and practical components are better suited to the operational requirements in an agricultural environment and that lower-cost designs meet farmers' needs for cost control. we try our best to simplify our institutional design to ensure the stability and reliability of our institutions and reduce the cost of the entire vehicle.

(2) Stable and efficient software

We are committed to designing stable, reliable, and efficient software to ensure the efficiency and accuracy of our identification and harvesting by using advanced and efficient algorithms.

4. Approach and Originality

4.1 Approach

4.1.1 Hardware

To meet the competition requirements for accurate cotton picking, the robot was controlled by a Jetson Nano in collaboration with an Arduino. To pick cotton in different directions and at different heights, we calculated the 3D spatial information obtained from the binocular camera to pinpoint the position of the cotton bolls and controlled the manipulator built with four high-torque servos to move the end-effector to the target position, achieving precise point-to-point picking. With our roller structure, the robot can pick cotton balls efficiently without damaging other plant tissues. The Jetson Nano is powered by a lithium battery with an output voltage of 16V via the UBEC module, and the Arduino and the electronic components with different operating voltages are powered by the voltage regulator module.

4.1.2 Software

We use the multi-threaded processing capability of the Jetson Nano to implement image recognition, target detection, kinematic solving of the manipulator, and serial communication with the Arduino. The Arduino receives the results from the Jetson Nano and controls the movement of the stepper motor, servo, and end-effector to complete the movement and cotton-picking operation, and display the cotton position on the LCD.

4.1.3 Conclusion of Approach

Overall, our robot controls all the electronic components on the robot through the Jetson Nano and Arduino working in tandem. the Jetson Nano connects to the binocular camera via the USB interface to process the captured images and use the recognition results to calculate the motion of the manipulator is communicated with Arduino through the UART serial port, and Arduino connects to the laser sensors to process the robot's position and attitude in the field and generate motion trajectories, drives stepper motors and servos to move and pick the robot based on the processing results.

4.2 Originality

4.2.1 Innovative Harvest Roller

We designed an innovative roller end-effector to achieve precise and harmless picking of cotton balls. This greatly reduces the complexity and difficulty of the algorithm, reduces the cost, and increases productivity.

4.2.2 Binocular Camera

Based on the principle of binocular parallax, a binocular camera uses imaging devices to acquire two images of the object under test from different positions and obtains three-dimensional information about the object by calculating the position deviation between the corresponding points of the images. Accordingly, the three-dimensional spatial position of each target cotton ball can be precisely located, laying the foundation for precise picking.

4.2.3 Multi-threaded Process

We use Jetson Nano as the processor and call Python's multi-threaded process module. Here we use three processes of Raspberry Pi to run simultaneously to fully utilize the memory and performance advantages of Jetson Nano to improve the processing speed.

4.2.4 Remote Emergency Brake

With the wireless transmission module and Bluetooth module of the Jetson Nano, we designed several remote emergency braking schemes. Although a start/brake switch is already available on the robot, considering that there is a certain safety risk of artificially approaching the brake when an unpredictable emergency occurs, we designed a remote emergency brake scheme where the switch is connected to the Bluetooth module, which in turn can control the operation of the Raspberry Pi; in addition, using WIFI wireless communication technology, we can access the Jetson

Nano's graphical interface to operate, stop the process of Raspberry Pi at any moment, and stop the process of Arduino through the communication program of Jetson Nano and Arduino, and then the robot operation.

5 Hardware Description of the Mechanical Structure

5.1 Overview

The overview of the robot is shown below, where Figure 5-1 and Figure 5-2 show the model picture, and Figure 5-3 and Figure 5-4 show the actual robot picture.

Figure 5-1

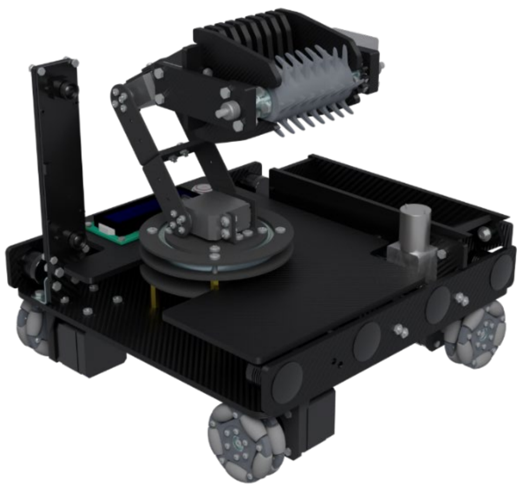


Figure 5-2

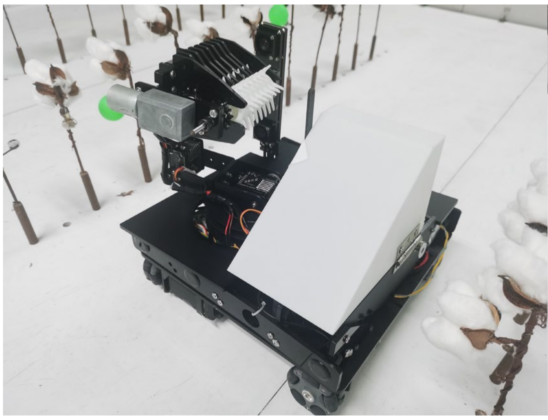


Figure 5-3

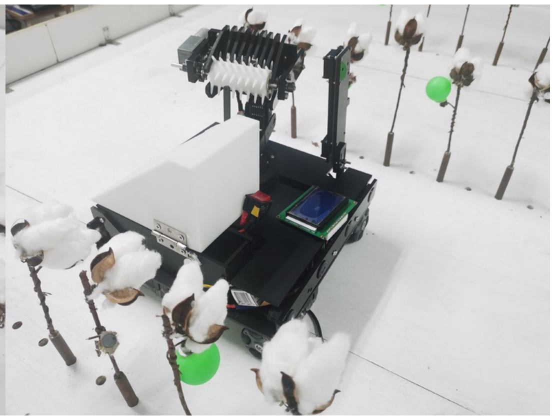


Figure 5-4

5.2 Omni wheel Structure

To make the robot more flexible and faster to move on the venue, and to better model the kinematics to achieve high-quality motion planning, we designed the walking mechanism using four omni wheels placed vertically on each other on the chassis, which makes our robot have good stability on the flat road.

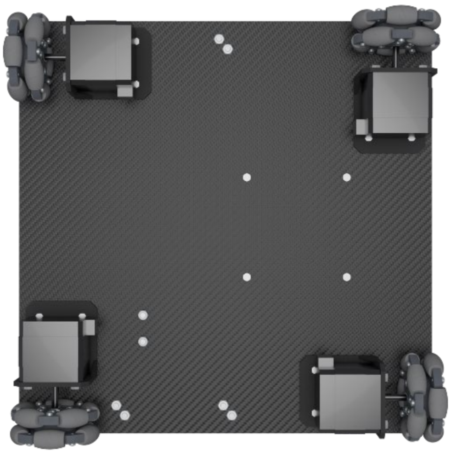
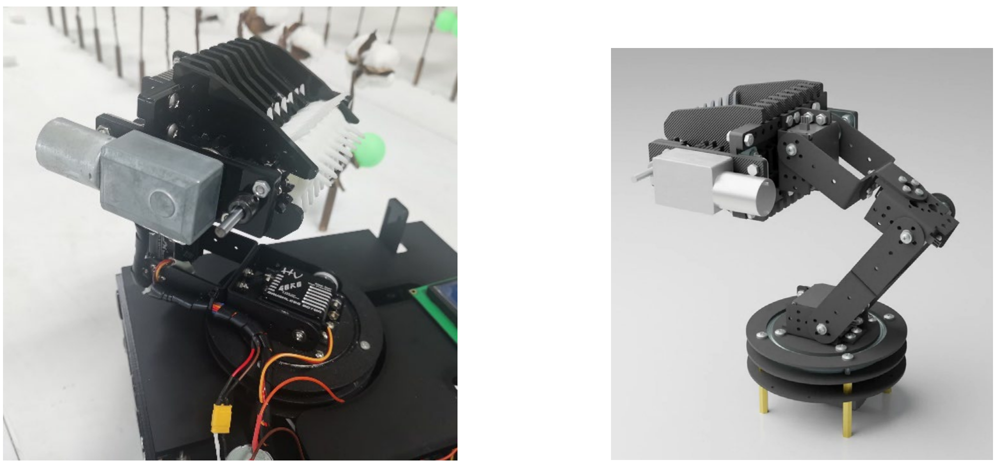


Figure 5-5 Omni wheel

5.3 Manipulator Structure

To save space and achieve flexible and personalized operation, we designed a manipulator consisting of four high-torque servos, as shown in Figure 5-6. Each servo can rotate 180°, thus forming the operating space of the manipulator, as shown in Figure 5-7. By setting the position and angle of different joints, the manipulator can be controlled to reach any position and any orientation in the operation space.



(a) Actual picture **(b) Model picture**

Figure 5-6 Manipulator structure

5.4 Harvest Roller Structure

Referring to the mature products in the market, our initial version of the end-effector consists of a roller and a baffle. The principle is to use the spikes of the drum to bite the cotton fibers to collect the cotton balls and use the baffle to block and drop the cotton balls , as shown in Figure 5-7. However, after actual experimentation, the collected cotton balls were easily blocked by the motor torque and the roller teeth.

Figure 5-7 Initial end-effector

Finally, we change the shape of the baffle to take advantage of the motor blocking phenomenon, using the strategy of collecting - blocking - fixing - storaging of cotton balls, which successfully improve the success of the operation while reducing the loss caused by the roller rubbing cotton balls.

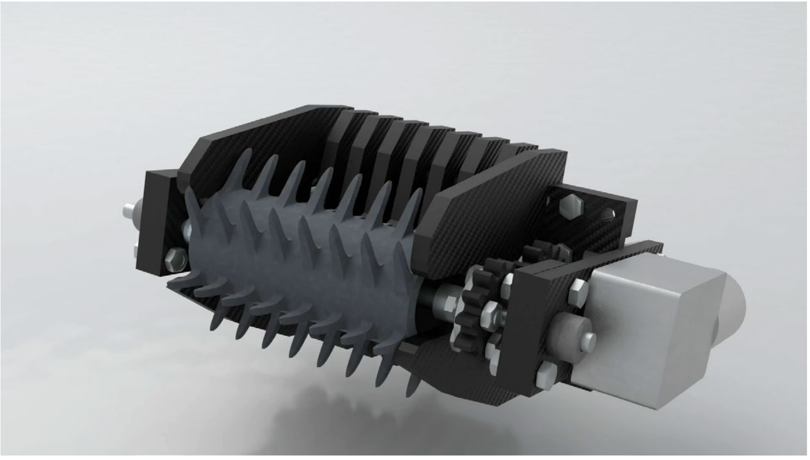
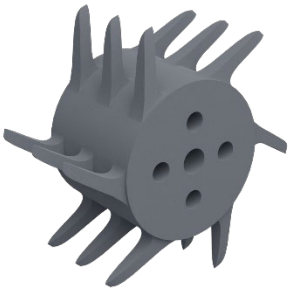
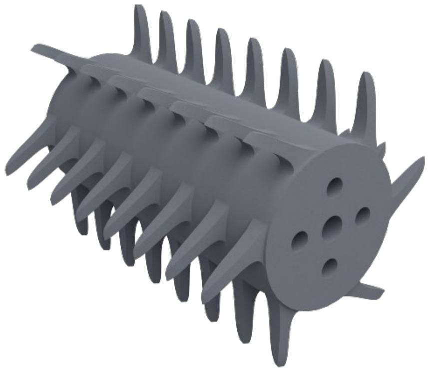


Figure 5-8 Final end-effector

On this basis, to improve the stability of the operation, we changed the small roller into a widened roller, the specific parameters are compared as shown in table 5-1.

Table 5-1 Roller Parameters

	Before	After
Model		
Dimension	56mm*54mm*25mm	56mm*54mm*75mm

In addition, to reduce the load on the servo of the manipulator, we used more buckle structures such as Figure 5-9 instead of metal connectors on the end-effector components.

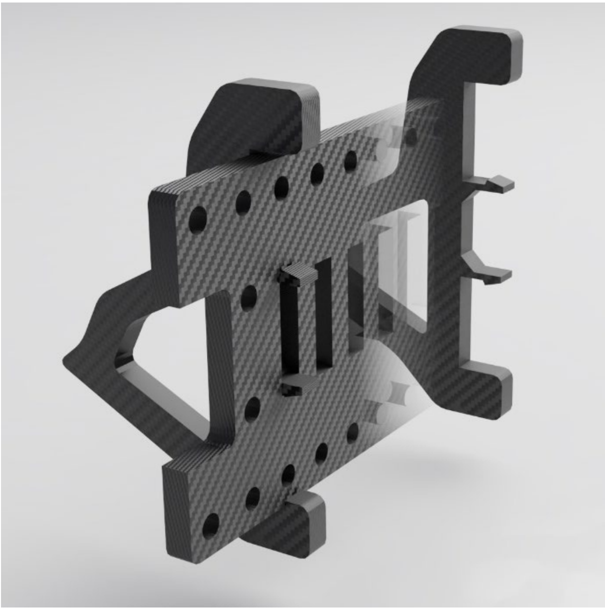


Figure 5-9 Plastic buckle structures

Overall, we achieved a higher success rate, stability, and low loss rate after several attempts to purposefully test and improve the cotton collection device, and the overall process is shown in Figure 5-10.

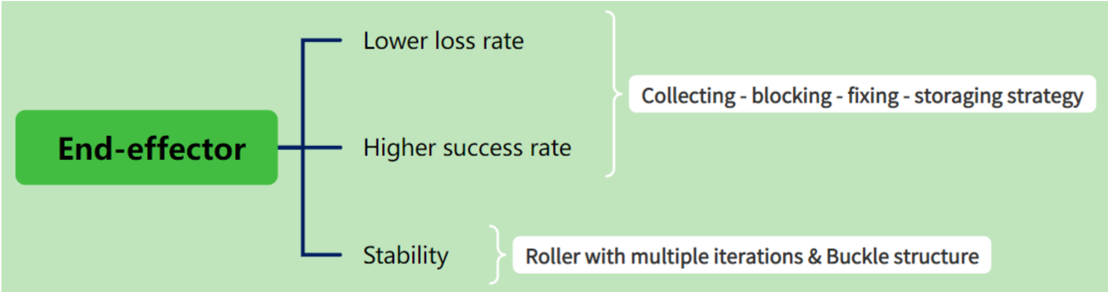


Figure 5-10 Overview of the end-effector

6 Hardware Description of the Main Electronic Components

6.1 Main Control Device

To improve the speed of image recognition and control the cost, we choose Jetson Nano to run the deep learning model image algorithm, supplemented by Arduino to control various sensors and power equipment. Jetson Nano developer kit has 128 CUDA cores, which can effectively handle parallel computing tasks, especially visual recognition tasks. NVIDIA Jetson series devices include Jetson Nano, Jetson TX1, Jetson TX2 series, Jetson NX, and Jetson AgX Xavier series. Among them, the Jetson Nano series is the cheapest and sufficient for our computing tasks. Jetson Nano has 2GB and 4GB versions. The official price of 2GB is $49, and the official price of the 4GB version is $99. The 2GB version is enough to meet our mission requirements, so we choose the 2GB version.

Arduino also has several versions, including Uno, Leonardo, mega and mega 2560, etc. Among them, the mega 2560 Pro has many pins and a small volume, which is very easy for users to start and install. To focus on solving practical problems, we choose Mega 2560 Pro as our MCU controller.

6.2 DC Motor

We choose the CHW-GW4632-370 model of miniature DC worm geared motor， it uses the whole gear material to provide more wear-resistant and longer service. And its good power-off self-locking force could guarantee smooth stretch ability and precise operation. The rotor part is made of the pure copper group with higher electrical conductivity , making the torque bigger to meet our requirements.

6.3 Laser Sensor

XKC-KL200-V laser sensor can be adjusted at will over a wide range of distances, which is 10-4000mm and can meet our requirements for the entire site distance detection. Laser sensors are precise and sensitive, so it’s our best choice.

6.4 Binocular Camera

HBV-1714-2 S2.0 is an excellent high-definition binocular camera. Its module size is only around 156mm × 36mm × 17.5mm, and the object distance can be 30CM to infinity, which is fairly suitable for our detection.

The Binocular Camera can measure the depth of the cotton balls, and then we can achieve the goal of precision positioning.

6.5 Steering Gear

JX HV7146MG is a perfect DC motor with Steel Gear Full CNC Aluminum Brushless Standard Servo. It is equipped with high voltage, high-performance programmable digital brushless standard servo as well as high-precision Taiwan-made steel gears with hard anodizing, which can guarantee the stable and precise movement of the robot.

6.6 Circuit Board

We designed a custom PCB board for connecting the Jetson Nano, Arduino, and other electronic components, the connection among the Jetson Nano, Arduino, and other electronic components is shown in Figure 6-1. (To show the connection between electronic components visually and clearly, only an abbreviated connection diagram is made here, not the actual connection situation)

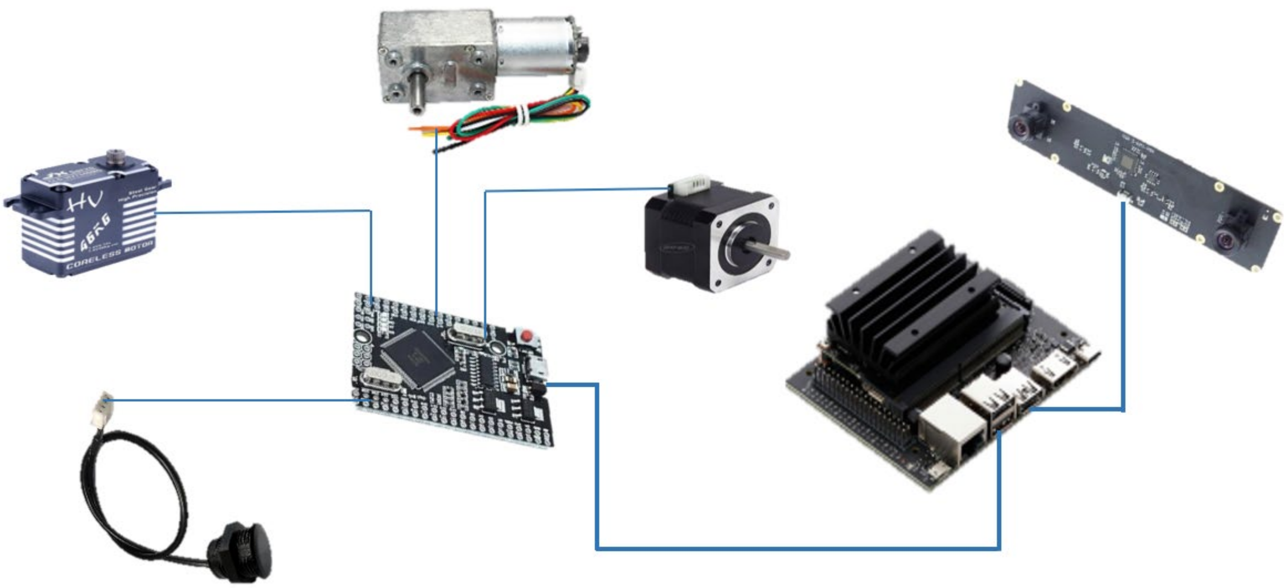


Figure 6-1 Schematic diagram of the connection of the electronic components

7 Software Overview and Logic Flowchart

7.1 Main Program Design

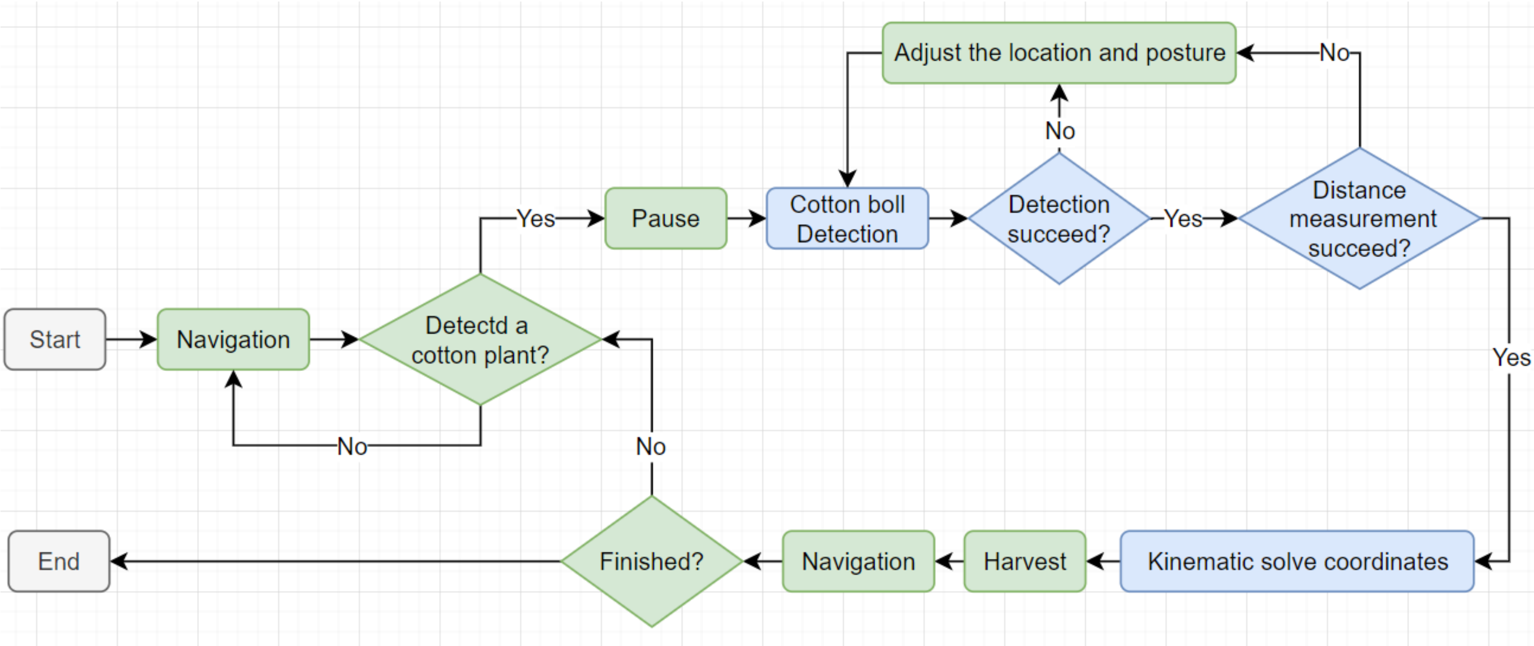


Figure 7-1 The Algorithm Flowchart

(The green and blue modules are executed by Arduino controller and Jetson Nano respectively)

Figure 7-1 above shows the flowchart of the algorithm, with the yellow part executed by Jetson Nano and the green part performed by Arduino. When the robot is in the start area, press the switch to start the robot. The Arduino controls the stepper motor to move the robot forward and controls the direction with the help of a laser sensor. When the laser sensor on the cotton side detects a cotton stalk, to prevent continuous detection, the robot will travel a short distance before stopping, meanwhile sending data to the Jetson Nano to open the camera for detection. If the cotton ball detection and distance measurement are unsuccessful, fine-tune the robot position and try again. If successful, collect the closest cotton ball coordinates, compute the servo angles, move the end-effector to the cotton ball, roll it down, gather it into the container, and finally transmit a signal to the Arduino to proceed walking. Repeat the above process until walking through the entire work area. When finished, the robot moves to the end area.

7.2 Tracking Design and Walking Route Map

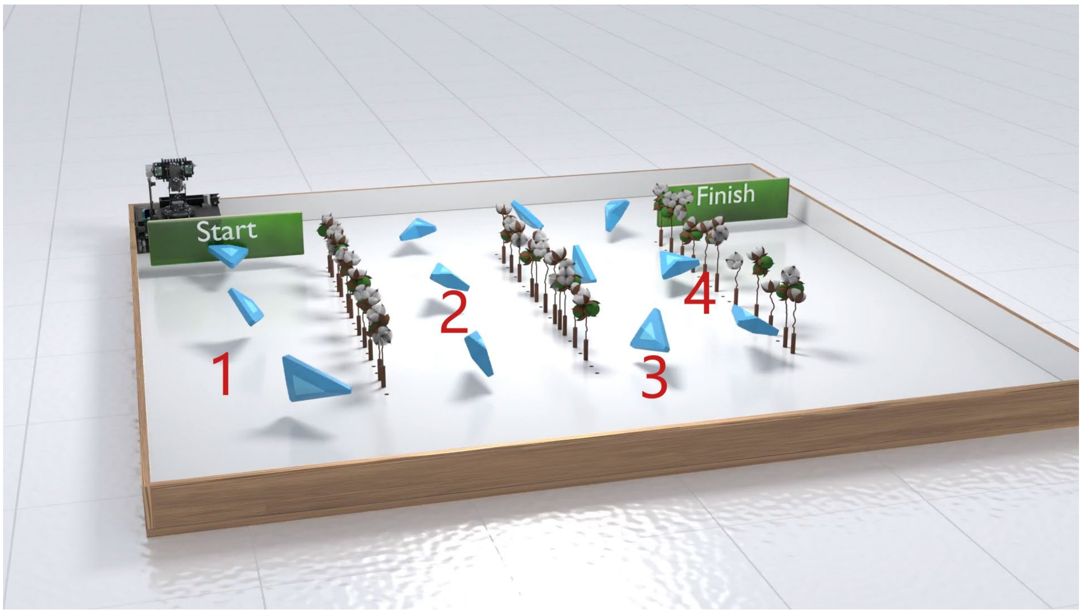


Figure 7-2 The walking route map of the robot

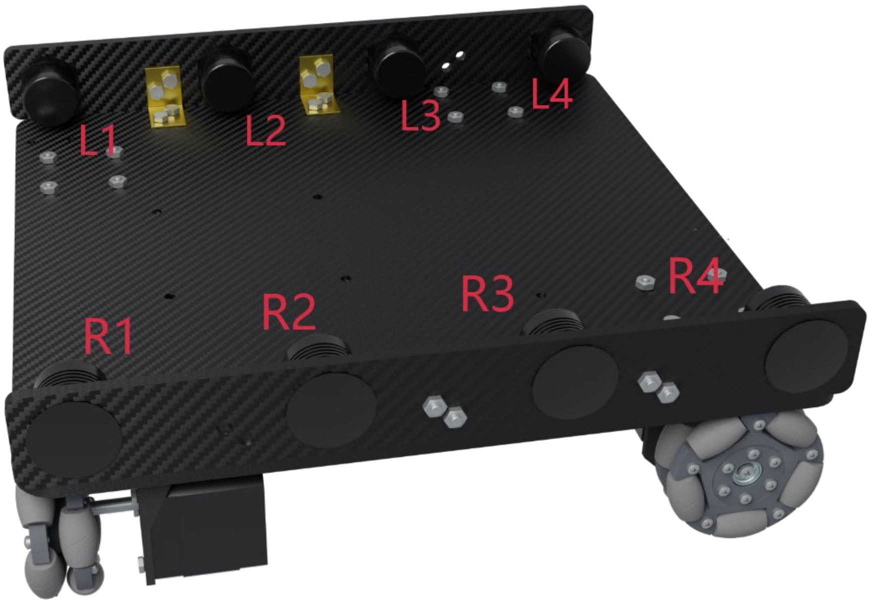


Figure 7-3 The distribution of onboard laser sensors

We use seven laser sensors to control the robot's movement. When walking on route ①, we measure the distance to the side plate on the right using R1 and R3, adjust the speed, and measure the number of cotton plants passed by on the left using L1. After L1 counts to eleven, we translate to the left at the end of route ① and enter route ②. In route ②, use R2 and R4 to measure the distance to the right side plate and adjust the speed, using L1 to measure the number of cotton plants passed by on the left. In route ③, use L2 and L4 to control the distance to the left side plate and regulate the speed, and use L1 to measure the number of cotton plants passed by the left side of the robot. After L1 counts to eleven, return to the route ④. In route ④, L2 and L4 are used to control the distance from the left plate and adjust the speed, and L1 is used to measure the number of cotton plants that the robot passes on the left side. After the L1 measurement counts reach eleven, the robot goes left to the end area.

7.3 Machine Vision

7.3.1 CNN & YOLO v5

Convolutional Neural Network (CNN) is a feed-forward neural network with artificial neurons that respond to continuous patches in certain feature spaces (called receptive field) and can be applied in areas such as speech recognition, image processing, and image recognition.

AlexNet topped the image classification task at imageNet 2012 with an error rate of 15.3%, marking the resurgence of neural networks and the rise of deep learning, proving that neural networks can shine in complex tasks like image recognition and processing.

In this competition, we used YOLO v5, a popular deep learning object detection, and segmentation framework.

Table 7-1 The performance of YOLO v5

Model	size(pixels)	mAP^{val} 0.5:0.95	mAP^{test} 0.5:0.95	mAP^{val} 0.5	Speed 100(ms)
YOLOv5s	640	36.7	36.7	55.4	2
YOLOv5m	640	44.5	44.5	63.1	2.7
YOLOv5l	640	48.2	48.2	66.9	3.8
YOLOv5x	640	50.4	50.4	68.8	6.1
YOLOv5s6	1280	43.3	43.3	61.9	4.3
YOLOv5m6	1280	50.5	50.5	68.7	8.4
YOLOv5l6	1280	53.4	53.4	71.1	12.3
YOLOv5x6	1280	54.4	54.4	72	22.4
YOLOv5x6TTA	1280	55	55	72	70.8

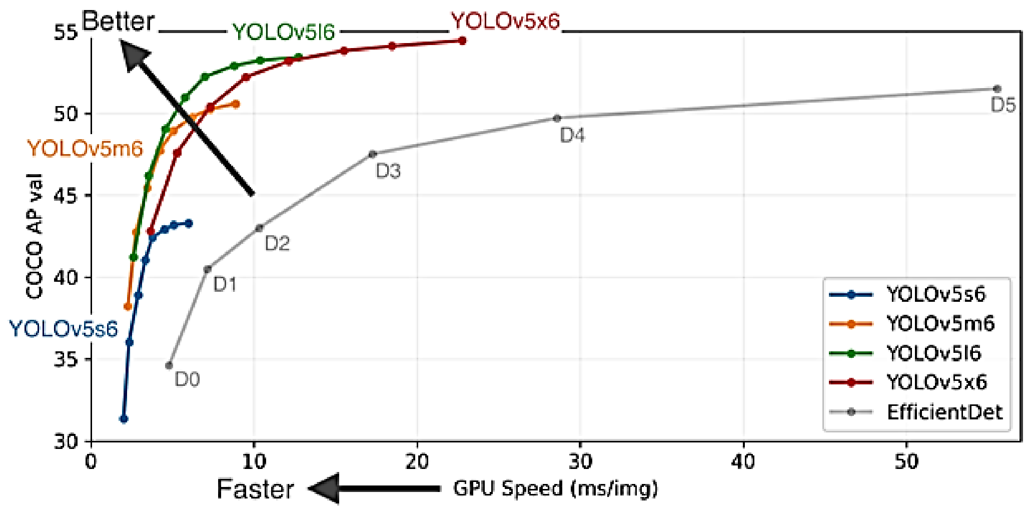


Figure 7-4 The performance of different YOLO v5 models on the COCO dataset

Among all the YOLOv5 networks, YOLOv5s is the smallest. The others are extended versions of it, with higher performance with the cost of inference speed. Considering the limited computation resource of the edge computing device Jetson

Nano and the accuracy required for object recognition in this competition, we choose to use the YOLOv5s model.

7.3.2 Dataset

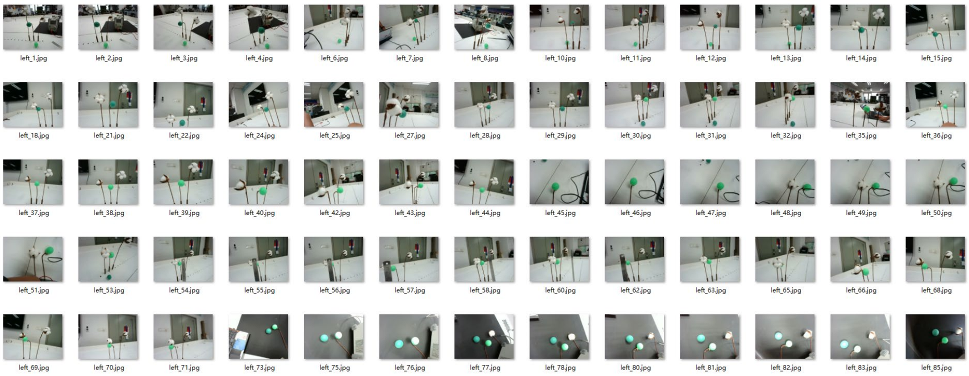


Figure 7-5 The raw images of the dataset

Both images of clean background and unstructured background are taken to build the dataset, thus tremendously improving the robustness of our model.

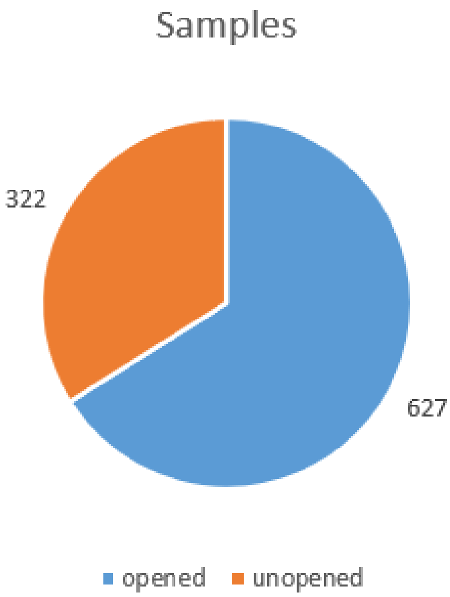


Figure 7-6 Sample num comparison

Regarding the ratio of samples from a different class, our dataset is composed of 322 unopened cottons and 627 opened cottons correspondingly.

7.3.3 Object Detection

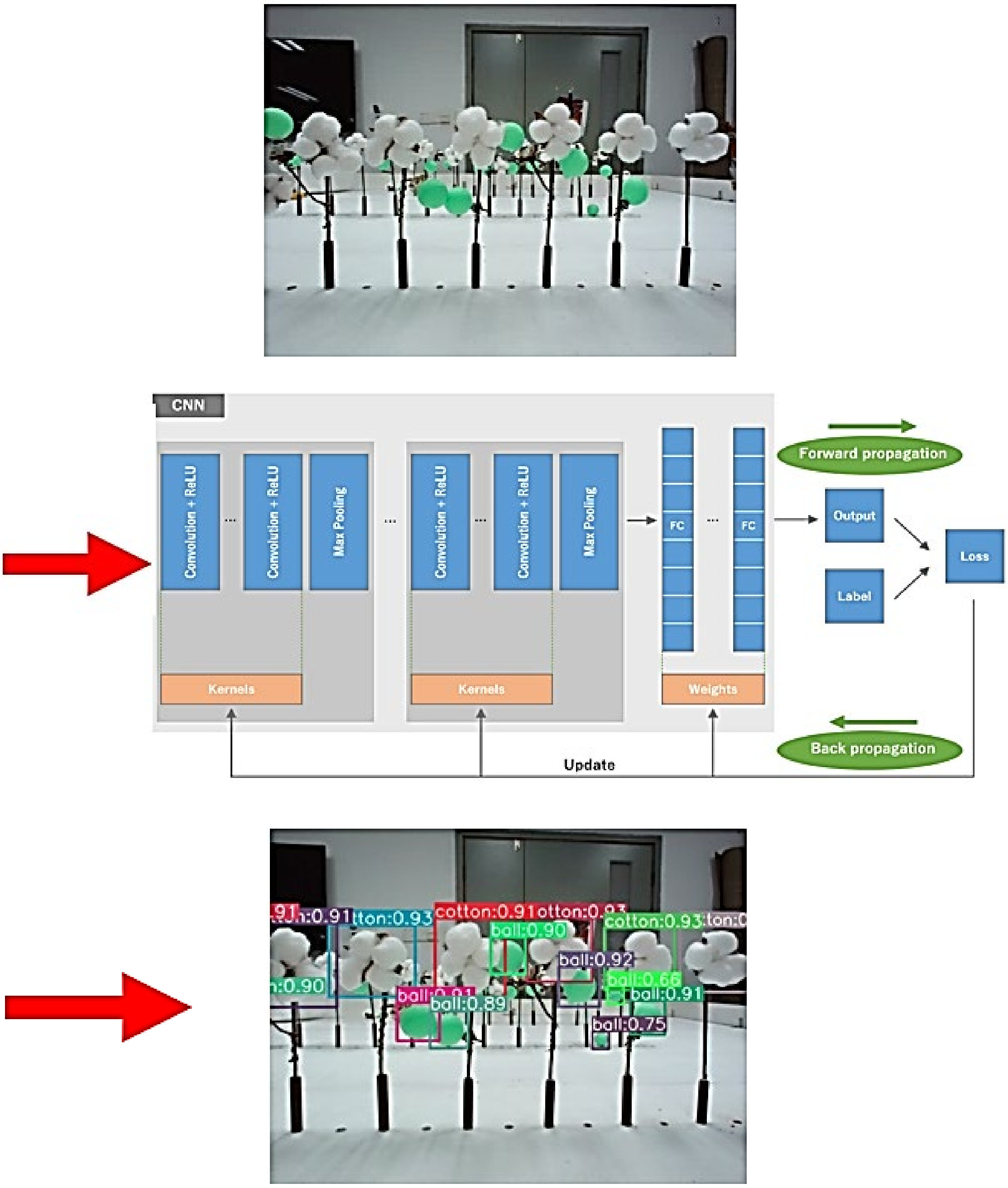


Figure 7-7 Process of detecting cottons

The figure above shows the process of detecting cottons in a photo. Apparently, our model is robust enough to correctly detect opened cottons and unopened cottons regardless of their distance, size, and pose.

7.3.4 Distance Measurement

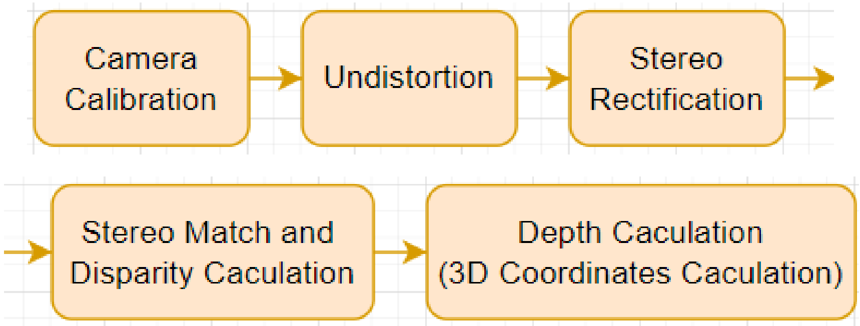


Figure 7-8 The pipeline for binocular depth estimation.

Light passing through the camera's optical system often fails to project onto the sensor as it should under ideal circumstances and produces distortions. In real-world situations, the camera lens often bend straight lines, and the closer you get to the edge of the image, the more significant this phenomenon becomes. In binocular stereo vision, to match the pixels of images taken by the two cameras, it is usually needed to remove the distortions and get the relative position between the two cameras. So we need to perform camera calibration, after which the poles of the two cameras are at infinity, the optical axes are parallel to each other, and the vanishing points are at the same height on the left and right images.

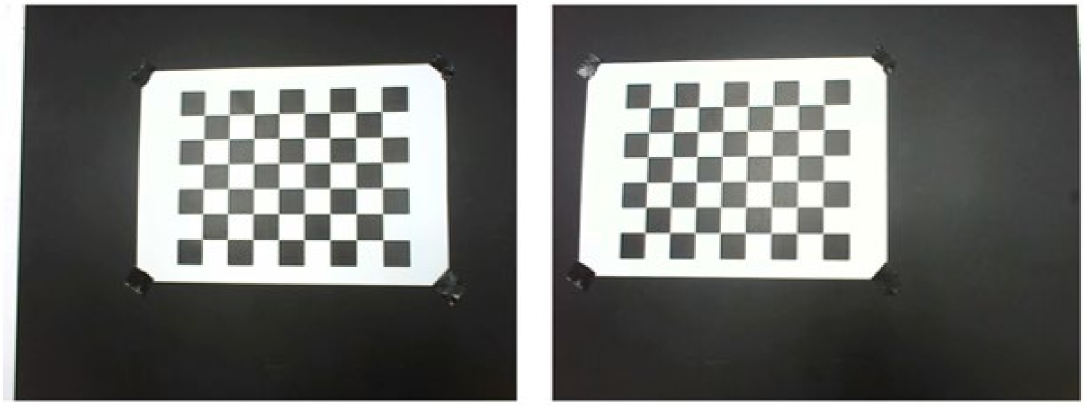


Figure 7-9 An example of calibrated images

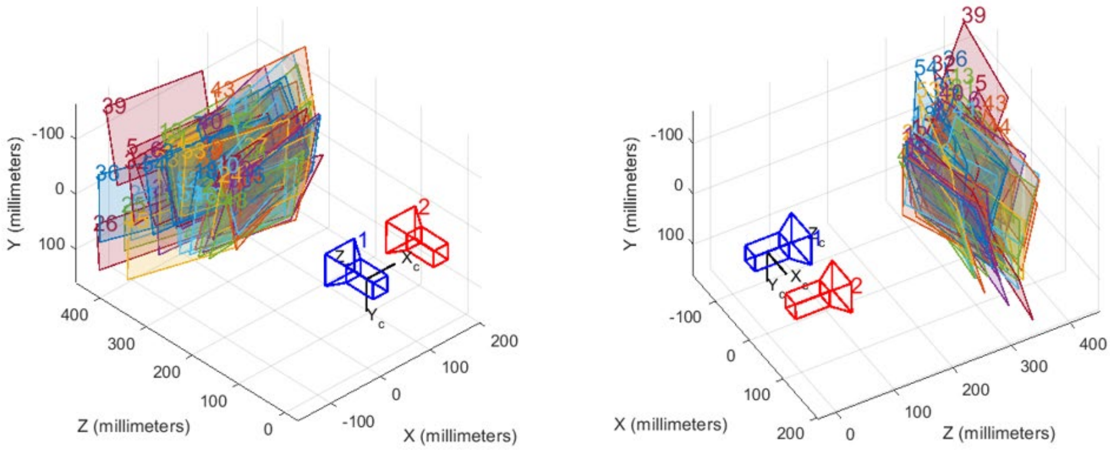


Figure 7-10 The schematic diagram of binocular calibration.

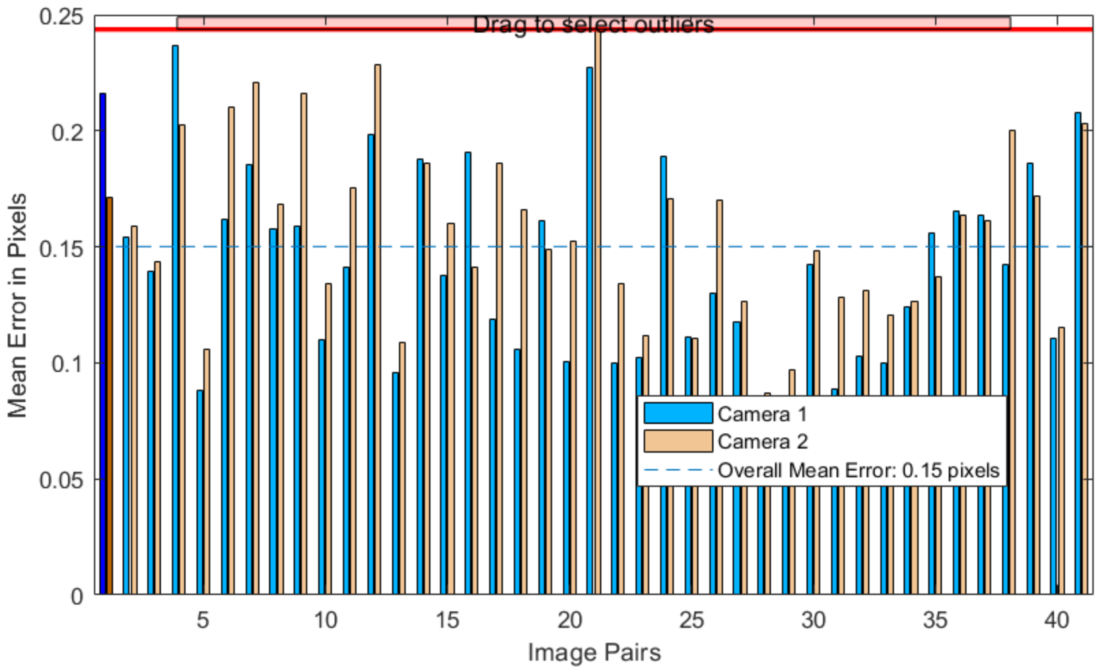


Figure 7-11 The results of binocular calibration

The average error of left and right image matching after calibration is less than 0.25, which means the calibration effect is good. Then, we obtain the camera's intrinsic matrix, extrinsic matrix, and distortion coefficients.

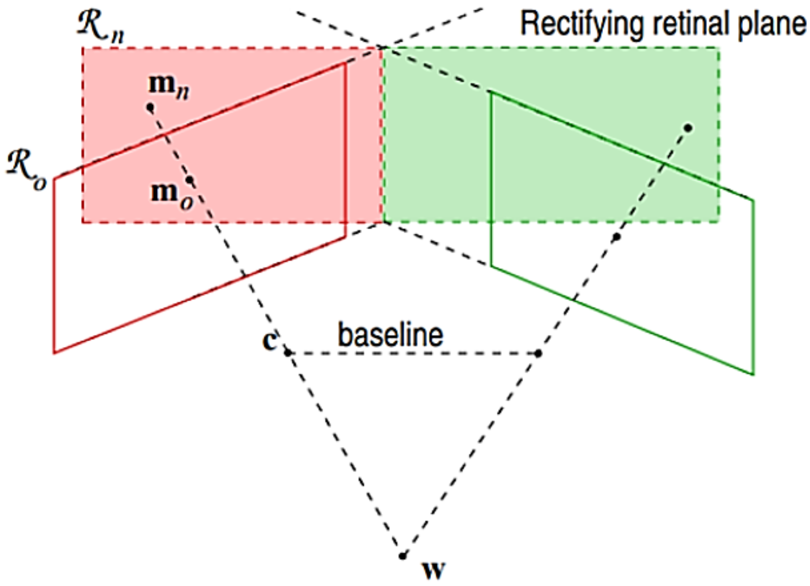


Figure 7-12 The schematic diagram of stereo correction.

We perform stereo correction after camera calibration to mathematically transform the left and right views of the same scene so that the two imaging planes are parallel to the baseline and the same point is located on the same line in both left and right images, referred to as coplanar row alignment.

This is the prerequisite for calculating the distance using geometry.

After stereo correction, stereo matching and disparity map calculation are performed. We use the SGBM algorithm to measure the disparity maps. Finally, we use the disparity maps and the reprojection matrix to convert the image coordinates into 3D coordinates to achieve 3D depth estimation.

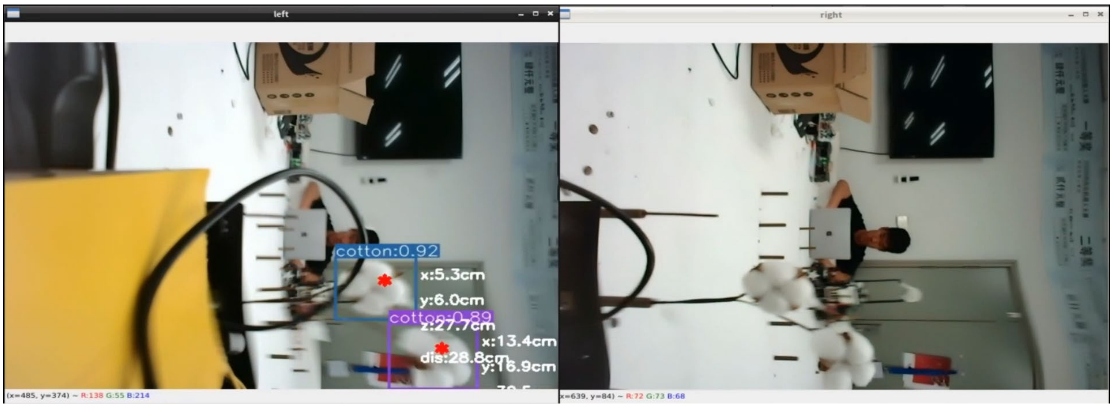


Figure 7-13 The results of stereoscopic depth estimation.

7.4 Harvest Control

7.4.1 3D Space Coordinate Conversion

To make sure the camera coordinate system (CCS) aligns with the robot coordinate system, we applied the following rotation on the CCS as follows:

$$\begin{cases} X'_{\text{Camera}} &= Y_{\text{Camera}} \\ Y'_{\text{Camera}} &= Z_{\text{Camera}} \\ Z'_{\text{Camera}} &= X_{\text{Camera}} \end{cases}$$

Once the coordinate systems are aligned, we performed hand-eye calibration to convert the 3D camera coordinates into the robotic arm coordinates.

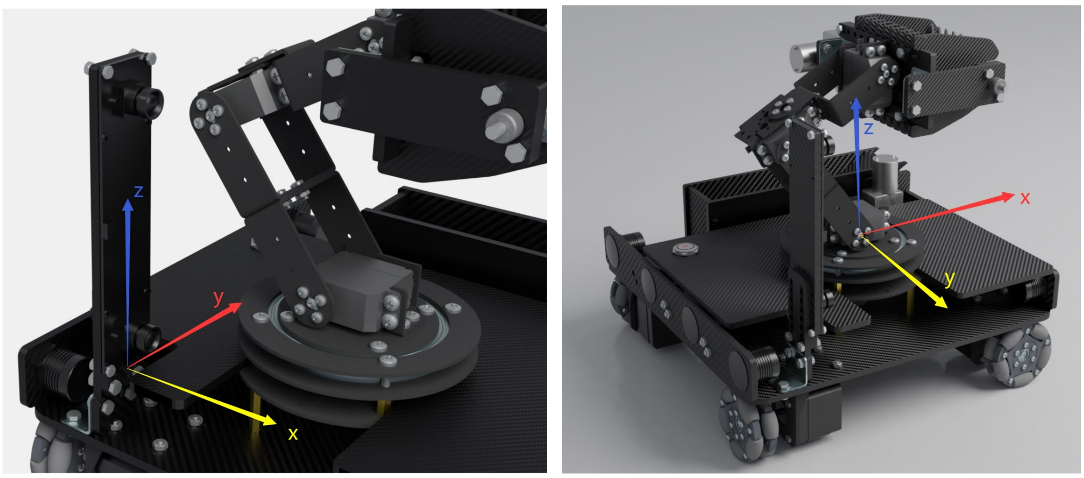


Figure 7-14 The camera coordinate system and the robot coordinate system

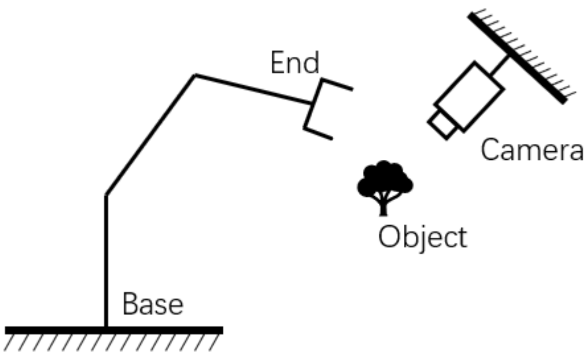


Figure 7-15 The schematic diagram of hand-eye calibration (eye to hand).

Since the positions of the robotic arm's end and the object are fixed, we have:

$$T_{\text{Base2 2}}^{\text{End}} \times T_{\text{Camera2}}^{\text{Base2}} \times T_{\text{object}}^{\text{Camera2}} = T_{\text{Base 1}}^{\text{End}} \times T_{\text{Camera1}}^{\text{Base 1}} \times T_{\text{Object}}^{\text{Camera1}}$$

$$\left(T_{\text{Base1}}^{\text{End}}\right)^{-1} \times T_{\text{Base2}}^{\text{End}} \times T_{\text{Cameras}}^{\text{Base}} = T_{\text{Camera1}}^{\text{Base1}} \times T_{\text{object}}^{\text{Camera1}} \times \left(T_{\text{Object}}^{\text{Camera2}}\right)^{-1}$$

Let

$$A = \left(T_{\text{Base 1}}^{\text{End}}\right)^{-1} \times T_{\text{Base 2}}^{End}$$

$$B = T_{\text{Object}}^{\text{Camera 1}} \times \left(T_{\text{Object}}^{\text{Camera 2}}\right)^{-1}$$

$$X = T_{\text{Cameras}}^{\text{Base 2}} = T_{\text{Camera 1}}^{\text{Base 1}}$$

The hand-eye calibration process solves the conversion matrix X between the camera and the robot arm base coordinate system. We use the nine-point calibration method to find this matrix.

7.4.2 Kinematic Solution

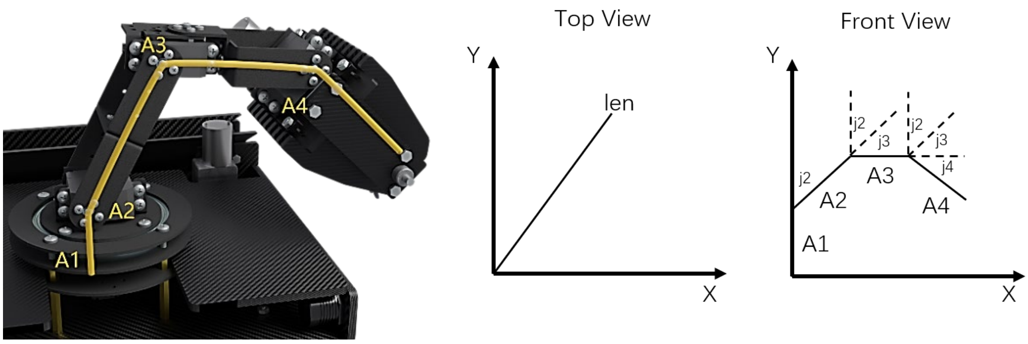


Figure 7-16 The mathematical model of the robot arm.

To calculate the forward kinematics, we have

$$\text{len} = A_2 \times S_2 + A_3 \times S_{2+3} + A_4 \times S_{2+3+4}$$

$$high = A_1 + A_2 \times C_2 + A_3 \times C_{2+3} + A_4 \times C_{2+3+4}$$

Where S refers to the sine and C represents the cosine.

$$Z = \text{high}$$

$$X = \text{len} \times C_1$$

$$Y = \text{len} \times S_1$$

The process of solving the inverse kinematics problem is shown below:

$$-\infty < x < +\infty$$

$$Y \geqslant 0$$

$$Z \geqslant 0$$

$$\text{len } = \sqrt{(Y+P)^2 + X^2}$$

$$high = Z$$

$$\begin{cases} \text{len } = 0 & \text{for any } j1 \\ \text{len } > 0 & \tan j_1 = Y/X, j_1 = \arctan(Y/X) \end{cases}$$

set

$$\alpha = j_2 + j_3 + j_4$$

then

$$\text{len } - A_4 \times S_\alpha = A_2 \times S_2 + A_3 \times S_{2+3}$$

$$\text{high } - A_1 - A_4 \times C_\alpha = A_2 \times C_2 + A_3 \times C_{2+3}$$

$$\text{set} \quad L = len - A_4 \times S_\alpha, H = high - A_1 - A_4 \times C_\alpha$$

then

$$\text{L } = A_2 \times S_2 + A_3 \times S_{2+3}$$

$$\text{H } = A_2 \times C_2 + A_3 \times C_{2+3}$$

$$L^2 + H^2 = A_2^2 + A_3^2 + 2 \times A_2 \times A_3 \times C_3$$

then

$$C_3 = \frac{L^2 + H^2 - A_2^2 - A_3^2}{2A_2A_3}$$

$$S_3 = \sqrt{1 - C_3^2}, \qquad j_3 = \arctan\frac{S_3}{C_3}$$

After transposition

$$L = S_2(A_2 + A_3C_3) + C_2A_3S_3$$

$$H = -S_2A_3S_3 + C_2(A_2 + A_3C_3)$$

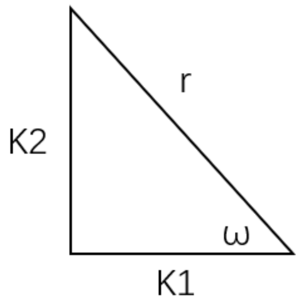
set

$$\begin{cases} A_2 + A_3 C_3 \\ A_3 S_3 \end{cases}$$

then

$$\begin{cases} L = K_1 S_2 + K_2 C_2 \\ H = K_1 C_2 - K_2 S_2 \end{cases}$$

K1, K2 can be considered as two sides of a right angle triangle.



$$\begin{cases} S_\omega = K_2/r \\ C_\omega = K_1/r \end{cases} \Rightarrow \omega = \arctan(K_2/K_1)$$

Divide both sides of (1) and (2) by r

$$\begin{cases} \frac{L}{r} = \frac{K_1}{r} S_2 + \frac{K_2}{r} C_2 = C_\omega S_2 + S_\omega C_2 = S_{\omega+2} \\ \frac{H}{r} = \frac{K_1}{r} C_2 - \frac{K_2}{r} S_2 = C_\omega C_2 - S_\omega S_2 = C_{\omega+2} \\ \qquad \Rightarrow \tan(\omega + 2) = \frac{S_{\omega+2}}{C_{\omega+2}} = \frac{L}{H} \\ \qquad \Rightarrow \omega + 2 = \arctan\frac{L}{H} \\ \qquad \Rightarrow j_2 = \arctan\frac{L}{H} - \omega \end{cases}$$

The final result is that a three-dimensional spatial coordinate is passed in and solved by the program into four servo angles to control the manipulator to move to the target position.

8 Parts List and Cost Analysis

As shown in Table 8-1, the total cost of our robot was \$623.07. As shown in Table 8-2, the total cost of computational devices to be integrated with our robot was \$67.36.

Table 8-1 Total cost

Material	Unit Price(USD)	Number	Total price(USD)
Acrylic board	/	/	5.97
Circuit board	1.49	1	1.49
Jetson Nano 2GB	59.00	1	59.00
Arduino Mega 2560 pro	8.36	1	8.36
Stepper motor	4.48	4	17.91
Motor drive	29.85	4	119.40
DC Motor	4.03	1	4.03
JX Servo BLS-HV7146MG	63.88	3	191.64
JX Servo BLS-HV7132MG	45.97	1	45.97
Laser Sensor	10.15	8	81.19
Binocular camera	32.69	1	32.69
3D printing consumables	/	/	5.97
Screws of different	/	90	0.30
Nuts of different	/	80	0.30
Lock nuts of M4	/	10	0.22
Angle iron	0.04	6	0.27
Bearing	0.45	4	1.79
Bearing support	0.45	4	1.79
Robotic arm	35.52	1	35.52
Motor support	0.37	4	1.49
Omnidirectional wheels	1.49	4	5.97
Switch and wires	/	/	1.79
Total			**623.07**

Table 8-2 Computational devices

Material	Unit Price (USD)	Number	Total price (USD)
Jetson Nano 2GB	59	1	59
Arduino Mega 2560 pro	8.4	1	8.4
Total			**67.36**

9 Reflection

There is no denying that our design still has a lot of immaturities, and we are constantly reflecting on our design and doing critical thinking.

9.1 Servo Control

For the control of the four servos that make up the manipulator, we did not achieve complete flexible control. To ensure the reliability of the mechanism, we reduced the speed of the servo rotation, but this made the overall movement speed of the manipulator decrease subsequently, which prolonged the robot operation time.

9.2 Exposed Components

Some of the electronic components of the robot are exposed on the outside and cannot cope with weather conditions such as rain.